Graph-based Clustering: Partitioning Data via Similarity Graphs

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

January 3, 2026

Abstract

Graph-based clustering treats data points as nodes in a graph and defines clusters as groups of nodes that are strongly connected to each other but weakly connected to the rest of the graph. This perspective is especially powerful when similarities between data points are not naturally expressed by Euclidean distances, or when data already comes in graph form such as social networks, citation networks, or web graphs [1]. The most widely used approach in this family is spectral clustering, which constructs a similarity graph, computes the graph Laplacian, and uses its eigenvectors to embed nodes in a low-dimensional space where standard clustering algorithms can be applied. This article introduces the core ideas of graph-based clustering, describes the construction of similarity graphs and Laplacians, presents the spectral clustering algorithm, and discusses its strengths, limitations, and typical applications.

Keywords: Graph-based Clustering, Spectral Clustering, Similarity Graph, Graph Laplacian, Normalized Cuts.

1 Introduction

Many clustering algorithms operate directly in the original feature space using distances such as Euclidean or cosine distance. However, in many applications the most natural representation of data is a *graph*: nodes represent entities (users, documents, proteins) and edges represent relationships or similarities (friendships, citations, interactions).

Graph-based clustering explicitly builds on this view. The data are represented as a weighted graph $G = (V, E)$, where each node in V is a data point and edge weights reflect pairwise similarities. Clustering then becomes the task of partitioning this graph into subsets of nodes that are densely connected internally and sparsely connected across subsets.

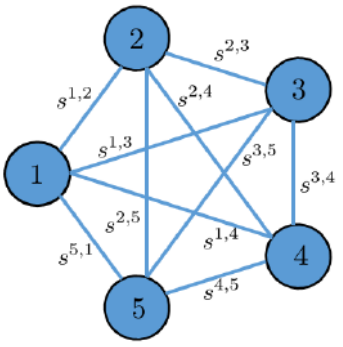

Figure 1: Example similarity graph: nodes are data points, edge thickness encodes similarity; graph-based clustering aims to cut weak connections while keeping strong ones.

Spectral clustering is the most common graph-based method: it leverages the eigenvectors of a graph Laplacian matrix to find an embedding of the nodes that reveals cluster structure. Compared to purely geometric methods like k-means, it can capture complex, non-convex cluster shapes.

2 From Data to Similarity Graph

2.1 Similarity Measures

Given a dataset $X = \{x_1, \ldots, x_n\}$, graph-based clustering first builds a **similarity matrix** $W \in \mathbb{R}^{n \times n}$, where W_{ij} measures how similar x_i and x_j are. Common choices include:

- **Gaussian (RBF) kernel** for continuous features:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right),$$

  where σ is a scale parameter.

- **Cosine similarity** for high-dimensional sparse features (e.g., text).
- **Graph adjacency** when the graph is given (e.g., social network edges).

In practice, W is often sparsified to reduce noise and computation:

- k-nearest neighbor graph: keep edges from i to its k nearest neighbors.
- ε-neighborhood graph: keep edges where distance $\leq \varepsilon$.

2.2 Degree Matrix and Graph Laplacian

Once the similarity matrix W is defined, the **degree matrix** D is the diagonal matrix whose entries are given by

$$D_{ii} = \sum_{j=1}^{n} W_{ij}.$$

The (unnormalized) **graph Laplacian** is defined as

$$L = D - W. \tag{1}$$

Two normalized variants are commonly used:

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}, \tag{2}$$

$$L_{\text{rw}} = D^{-1} L = I - D^{-1} W. \tag{3}$$

Illustrative example. Consider a simple weighted graph with three nodes whose similarity matrix is

$$W = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The corresponding degree matrix is

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The unnormalized graph Laplacian is therefore

$$L = D - W = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

This matrix encodes local connectivity information: diagonal entries reflect node degrees, while off-diagonal entries indicate pairwise connections between neighboring nodes.

The eigenvectors of the Laplacian (or its normalized variants) capture important structural properties of the graph and form the basis of spectral clustering methods.

3 Spectral Clustering

3.1 Intuition: Cuts and Balanced Partitions

A natural way to cluster a graph is to cut edges so that connections between clusters are few or weak, while connections inside clusters remain strong. One simple measure is the **cut** between disjoint subsets A and B:

$$\mathrm{cut}(A, B) = \sum_{i \in A} \sum_{j \in B} W_{ij}.$$

Minimizing the cut alone tends to isolate small sets of nodes. To avoid trivial solutions, normalized criteria such as **RatioCut** or **Normalized Cut (Ncut)** are used:

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(A)} + \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(B)},$$

where $\mathrm{vol}(A) = \sum_{i \in A} D_{ii}$ is the volume of A.

Finding the exact minimum Ncut is NP-hard, but a relaxed version leads to an eigenvalue problem involving the graph Laplacian. This is the basis of spectral clustering.

3.2 Algorithm (Normalized Spectral Clustering)

One popular version (Ng–Jordan–Weiss algorithm) for K clusters:

1. Construct a similarity graph and compute W.
2. Compute degree matrix D and normalized Laplacian $L_{\mathrm{sym}} = I - D^{-1/2} W D^{-1/2}$.
3. Compute the first K eigenvectors $u_1, \ldots, u_K$ of L_{sym} associated with the smallest eigenvalues.
4. Form matrix $U \in \mathbb{R}^{n \times K}$ with columns $u_1, \ldots, u_K$.
5. Normalize each row of U to unit length.
6. Treat each row of U as a point in $\mathbb{R}^K$ and cluster them into K groups using k-means.
7. Assign original points x_i to clusters according to the cluster of the i-th row of U.

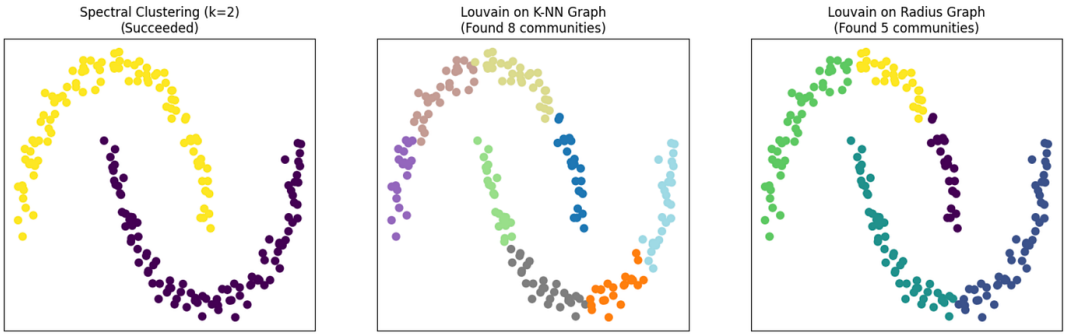

Figure 2: Spectral clustering on a two-moons dataset

The spectral embedding given by rows of U can be seen as a low-dimensional representation of the graph that preserves cluster structure.

4 Advantages, Limitations, and Variants

4.1 Advantages of Graph-based and Spectral Clustering

Graph-based clustering, and spectral clustering in particular, offers several benefits:

- **Non-convex clusters:** Can separate complex shapes (e.g., moons, rings) that defeat k-means in the original space.
- **Flexibility of similarity:** Works with any similarity measure that can be encoded in a graph, including non-metric or domain-specific similarities.
- **Natural for graph data:** Directly applicable when the input is already a graph (social networks, citation graphs, etc.).

4.2 Limitations

However, there are important challenges:

- **Scalability:** Computing eigenvectors of an $n \times n$ Laplacian scales poorly for very large n, although sparse methods and approximations help.
- **Parameter sensitivity:** Results depend on choices for similarity function, graph construction (e.g., k in k-NN graph), and number of clusters K.
- **Interpretability:** Eigenvectors are abstract; understanding clusters purely in terms of original features may require additional analysis.

4.3 Variants and Related Approaches

Several variants and related methods exist:

- **Unnormalized vs normalized Laplacians:** Different theoretical properties and behaviors when degrees vary strongly.
- **Multi-way cuts:** Extensions of Ncut or RatioCut to more than two clusters directly via multiple eigenvectors.
- **Graph Cuts in vision:** Energy-minimization formulations (e.g., min-cut/max-flow) widely used in image segmentation.

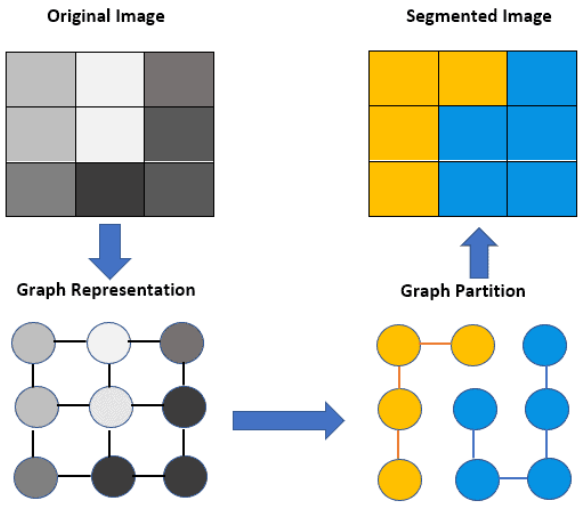

Figure 3: Graph-based image segmentation: pixels are nodes, edges encode similarity in color and proximity

5 Applications of Graph-based Clustering

Graph-based clustering is particularly attractive in domains where relationships are more natural than coordinates:

- **Social networks:** Detecting communities of users with dense friendship or interaction patterns.
- **Document and citation networks:** Grouping papers or web pages into topic communities based on hyperlinks or citation patterns.
- **Image segmentation and vision:** Treating pixels or superpixels as nodes and using graph cuts or spectral methods for segmentation.
- **Manifold learning:** Using neighborhood graphs to capture the intrinsic geometry of data lying on low-dimensional manifolds.

In many of these cases, graph-based clustering is used together with other techniques: for example, constructing a k-NN graph on learned embeddings (from autoencoders or transformers) and then applying spectral clustering to those embeddings.

6 Conclusion

Graph-based clustering shifts the clustering problem from raw feature space to the language of graphs and spectral analysis. By encoding pairwise similarities in a graph and analyzing the graph Laplacian, spectral methods can uncover complex cluster structures that are difficult to detect with purely geometric or density-based algorithms.

While eigen-decomposition and graph construction can be computationally demanding on very large datasets, approximate methods and sparse representations make graph-based clustering practical in many modern applications, from community detection in networks to image segmentation. Understanding how to build good similarity graphs and interpret spectral embeddings is key to making these techniques effective in practice.

References

[1] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.